IN THE UNITED STATES DISTRICT COURT

FOR THE EASTERN DISTRICT OF TEXAS

MARSHALL DIVISION

INTERGRAPH CORPORATION §

Vs. § CIVIL ACTION NO. 2:01CV160 -TJW

INTEL CORPORATION §

FINDINGS OF FACT AND CONCLUSIONS OF LAW

On July 2, 2002, came on for trial before the court, sitting non-jury, the above-entitled and numbered cause. At the conclusion of the evidence, the court directed the parties to file written briefs in the form of arguments. The court then heard oral closing arguments on August 29, 2002. After considering the admissible evidence, the applicable law and the arguments of counsel, the court issues these findings of fact and conclusions of law pursuant to Fed. R. Civ. P. 52. Any finding of fact that is actually a conclusion of law should be treated as a conclusion of law. Any conclusion of law that is actually a finding of fact should be treated as a finding of fact.

FINDINGS OF FACT

1. This is a patent infringement suit in which the plaintiffs charge the defendant with infringement of two United States Patents. The patents in suit are U.S. Patent Nos. 5,560,028 ('028 Patent) and 5,794,003 ('003 Patent).

2. The claimed inventions disclosed by the two patents relate to a computer architecture in which individual instructions may be executed in parallel by multiple processing pipelines and to

184

methods for executing individual instructions in parallel.

3. Intergraph Hardware Technologies Company, Inc. is the assignee of the '028 patent, granted to inventors Howard G. Sachs and Siamak Arya. The '028 patent is entitled "Software Scheduled Superscalar Computer Architecture." It issued September 24, 1996, based on an initial application filed November 5, 1993. The '028 patent has 23 claims. The asserted claims are Claims 6, 7, 8, 9, 11, 16, 20, and 21. Claims 6, 11, 16, and 20 are independent claims.

4. The abstract of the '028 patent provides an introduction to the patent and its field of technology:

> A computing system is described in which groups of individual instructions are executable in parallel by processing pipelines, and instructions to be executed in parallel by different pipelines are supplied to the pipelines simultaneously. During compilation of the instructions those which can be executed in parallel are identified. The system includes a register for storing an arbitrary number of the instructions to be executed. The instructions to be executed are tagged with pipeline identification tags and group identification tags indicative of the pipeline to which they should be dispached, and the group of instructions which may be dispatched during the same operation. The pipeline and group identification tags are used to dispatch the appropriate groups of instructions simultaneously to the differing pipelines. '028 Patent, Abstract.

5. Intergraph Corporation is the assignee of the '003 patent, granted to inventor Howard G. Sachs. The '003 patent is entitled "An Instruction Cache Associative Crossbar Switch System." The '003 patent issued August 11, 1998, based on continuations of an application initially filed November 5, 1993. The '003 patent has 33 claims. The asserted claims are Claims 1, 6, 7, 8, 11, 14, 15, 16, 17, 22, 28, and 33. Claims 1, 6, 8, 22, and 28 are independent claims.

6. The abstract of the '003 patent provides an introduction to this invention and its associated field of technology:

> A computing system as described in which individual instructions are executable in

> parallel by processing pipelines, and instructions to be executed in parallel by different pipelines are supplied to the pipelines simultaneously. The system includes storage for storing an arbitrary number of the instructions to be executed. The instructions to be executed are tagged with pipeline identification tags indicative of the pipeline to which they should be dispatched. The pipeline identification tags are supplied to a system which controls a crossbar switch, enabling the tags to be used to control the switch and supply the appropriate instructions simultaneously to the differing pipelines. '003 patent, Abstract.

7. The plaintiffs (collectively referred to in these Findings of Fact and Conclusions of Law as "Intergraph") have accused the defendant's Itanium and Itanium 2 processors of infringement in this action. The parties sometimes refer to the Itanium processor as the Merced processor and the Itanium 2 processor as the McKinley processor.

8. In the Itanium and Itanium 2 processors, computer instructions are organized into 128-bit bundles. Each bundle contains three syllables and a 5-bit template field that contains instruction information for each of the three syllables.

9. Hardware included in the Itanium and Itanium 2 processors uses the information provided in the template, along with other information, to decide which of the instructions in a bundle to execute in parallel. In addition, hardware in the Itanium and Itanium 2 processors uses the template information, along with other information, to route instructions in the bundle to appropriate processing pipelines for execution.

Infringement Analysis

10. The court applies its claim constructions ordered on June 3, 2002 and in the conclusions of law section of these Findings of Fact and Conclusions of Law.

11. Neither party requested the court to construe the word "instruction" as it appears in the patents in suit. For the reasons expressed in the court's conclusions of law, the court has construed

the word "instruction" to mean "the smallest unit of work capable of being processed at any particular stage in the computer."

'028 Patent

12. In its claim construction order of June 3, 2002, the court construed the term "group identifier" to mean "a designation indicative of a group of individual instructions . . . ." The court construed the term "groups/sets of individual instructions" as "a collection of one or more individual instructions that can be executed simultaneously (i.e. can be dispatched to processing pipelines simultaneously)."

13. The Itanium and Itanium 2 processors do not utilize "group identifiers" as the court has construed the terms.

14. Intergraph urges that "stop bits" formed by the compiler in the Itanium and Itanium 2 processors are group identifiers because they "demark the boundaries of the group which are encoded in the template." (Pls.' Post Trial Brief, p. 21-22). The court disagrees.

15. The stop bits indicate certain types of data dependencies. The stop bits are not "a designation indicative of a group of individual instructions."

16. Intergraph offered no evidence of infringement under the doctrine of equivalents.

17. Alternatively, the evidence does not persuade the court by a preponderance of the evidence that Intergraph has satisfied the court's definition of "group identifier" under the doctrine of equivalents.

18. Independent claims 6, 11 and 16 of the '028 Patent require the element of a group identifier.

19. Independent claim 20 provides:

> 20. In a computing system having a plurality of processing pipelines in which groups

> of individual instructions are executable, each individual instruction in a group executable in parallel by the plurality of processing pipelines, a method for transferring each individual instruction in a group to be executed through a crossbar switch having a first set of connectors coupled to a very long instruction word storage for receiving individual instructions therefrom, a second set of connectors coupled to the plurality of processing pipelines, and switches between the first set and the second set Of [sic] connectors, the method comprising:
>
> retrieving the very long instruction word from a main memory;
>
> storing in the very long instruction word storage, the very long instruction word, the very long instruction word having a set of individual instructions including at least one group of individual instructions to be executed in parallel, each individual instruction in the at least one group having embedded therein a unique pipeline identifier indicative of the processing pipeline which will execute that individual instruction, the very long instruction word storage also including at least one other individual instruction not in the at least one group of individual instructions, the at least one other individual instruction having embedded therein a different pipeline identifier; and
>
> using the unique pipeline identifiers of the individual instructions in the at least one group of individual instructions to control the switches between the first set of connectors and the second set of connectors to thereby supply each individual instruction in the at least one group to be executed in parallel to an appropriate processing pipeline. '028 patent, Claim 20.

20. Each limitation of claim 20 of the '028 patent is found in operation of the accused devices.

21. Dr. Bernard Peuto testified that, in his opinion, all of the limitations of all of the asserted claims were found in the accused devices. His infringement analysis during his trial testimony focused extensively on certain specific claims of the '028 patent, and, in addition to that testimony, he referred the court to the claim charts submitted by Intergraph to reflect his analysis as to the remaining asserted claims. As a number of the claims share common limitations, the court has reviewed Dr. Peuto's analysis carefully and has incorporated that analysis into the court's decisions with respect to the limitations of claim 20 that are common to the other claims of the '028 patent.

22. The accused processors are computing systems that contain a plurality of processing

pipelines, in which groups of instructions are executable.

23. Each of the instructions in a group is executable in parallel. The groups can vary in size from one instruction to several instructions or more.

24. The accused processors perform a method for transferring each instruction in the group through a crossbar switch having first and second sets of connectors.

25. The first set of connectors is coupled to a very long instruction word storage. "Coupled" means connected, directly or indirectly. The accused devices have a first set of connectors. That first set of connectors is between the crossbar switch and presentation and template identification latches.

26. The rotate buffer in the accused processors is a very long instruction word storage that is connected, directly or indirectly, to the crossbar switch. Therefore, the accused devices have a very long instruction word storage that is coupled to a crossbar switch through a first set of connectors.

27. The accused processors have a second set of connectors that are coupled to the plurality of processing pipelines.

28. The processing pipelines in the accused devices are the execution units.

29. The accused processors have switches between the first and second set of connectors. The switches are the steering crossbar in the Itanium processor and the multiplexers between the presentation latches and the issue ports in the Itanium 2 processor.

30. The accused processors retrieve the very long instruction word from a main memory. The main memory is the external memory from which the processors retrieve the very long instruction word.

31. The accused processors store the very long instruction word in the rotate buffer.

32. The very long instruction word that is retrieved contains a group of individual instructions.

33. Each instruction in the group has a pipeline identifier embedded in the instruction.

34. The pipeline identifier is indicative of the processing pipeline which will execute that instruction.

35. Both parties agreed that, at this stage of the processing, the template information is a part of the instruction and that a syllable is not a complete instruction. Because the template information contains the pipeline identifier, and because the template information is introduced as or an integral part of the instruction, the pipeline identifier is embedded in the instruction.

36. In many instances in the accused processors, within the rotate buffer, there will be a group of instructions plus at least one other instruction that is not in that group. Among many others, template 03 in Exhibit 111 would be such an example.

37. The pipeline identifiers in the template information control the steering crossbar (Itanium) and the multiplexers (Itanium 2) to supply each individual instruction to the appropriate processing pipelines.

38. Therefore, all limitations of claim 20 of the '028 patent are found in the accused processors.

39. Dr. Peuto testified he did not rely on the doctrine of equivalents. The court has not undertaken analysis of the doctrine of equivalents as to claim 20 of the '028 patent as Intergraph has not offered evidence or analysis under the doctrine of equivalents.

40. Claim 21 depends from claim 20. Claim 21 provides:

> 21. A method as in claim 20 wherein the step of using the pipeline identifiers comprises:
>
> supplying the unique pipeline identifiers of each individual instructions [sic] in the at least one group of individual instructions to individual decoders of a set of

> decoders, each decoder of which provides an output signal indicative of the Unique [sic] pipeline identifiers of the individual instruction supplied thereto; and
>
> using the output signals of the sets of decoders to control the switches between the first set of connectors and the second set of connectors to thereby supply each individual instruction in the at least one group to be executed in parallel to an appropriate processing pipeline.

41. The dispersal logic, decode and valid vector logic (Itanium) and the instruction syllable dispersal unit (Itanium 2) receive the pipeline identifiers and produce signals to control the steering crossbar (Itanium) and the multiplexers (Itanium 2). *See* Pl. Exh. 103 (Itanium) and Exh. 120, 124, 129 (Itanium 2).

42. For the reasons expressed in the preceding findings of fact and the court's findings of fact related to the limitations of claim 20, all limitations of claim 21 are found in the accused processors.

'003 Patent

43. Of the '003 patent, Intergraph asserts independent claims 1, 6, 8, 22, and 28.

44. Independent claim 1 provides:

> 1. A computing system comprising:
>
> means for forming groups of software-scheduled instructions, software-scheduled instructions within each of the groups executable in parallel; and
>
> a super-scaler cache for routing each of the software-scheduled instructions within the groups to be executed in parallel to an appropriate instruction pipeline of a plurality of instruction pipelines, the super-scaler cache comprising:
>
> super-scaler storage for holding one group of the groups of software-scheduled instructions, each software-scheduled instruction within the one group having embedded therein an instruction pipeline identifier of a plurality of instruction pipeline identifiers;
>
> an associative crossbar having a first set of connecters coupled to the super-scaler storage for receiving each of the software-scheduled instructions therefrom, and a second set of connectors coupled to the plurality of instruction pipelines; and

means responsive to the instruction pipeline identifier of each of the software-scheduled instructions, for coupling appropriate connectors of the first set of connectors to appropriate connectors of the second set of connectors, to thereby supply each of the software-scheduled instructions to the appropriate instruction pipeline for parallel execution.

45. Intel urged at trial that the accused processors did not infringe any of the asserted claims of the '003 patent because the accused processors did not have an associative crossbar physically located "in the cache." The court rejects that argument as the claims do not call for the limitation of having an associative crossbar physically located in the cache. To be sure, the claims call for a super-scaler cache consisting of certain elements, one of which is an associative crossbar. The claim limitations, however, do not expressly require the associative crossbar to be located physically within the cache.

46. Intel relied on extrinsic evidence in the form of an inventor drawing to support its argument. But Intel does not explain what it means for the associative crossbar to be "in" the cache. In any event, the inventor drawing does not support Intel's argument, as the drawing shows a cache and, separately, an associative crossbar. Finally, the court rejects the inventor drawing in favor of the intrinsic evidence. Specifically, none of the figures in the '003 patent shows a crossbar physically located within a cache. Therefore, the claim elements may be satisfied if Intergraph has proved the constituent requirements of the super-scaler cache limitation.

47. Intel also urged that its crossbar is not "associative." The court disagrees. The court defined "associative" to mean the data being routed includes the routing instructions. In the accused processors, the instructions being routed include the template information. The template information includes the routing instructions. These instructions include the information needed for routing. Therefore, in the Itanium processor, the steering crossbar is "associative" as defined by the court.

In the Itanium 2 processor, the multiplexers are "associative" as defined by the court.

48. The accused Itanium processor has an associative crossbar consisting of the steering crossbar and the associated dispersal logic.

49. The accused Itanium 2 processor has an associative crossbar consisting of the multiplexers and associated dispersal logic.

50. All limitations of claim 1 are found in the accused processors.

51. The accused processors are computing systems. Each of the processors utilizes a compiler for forming groups of instructions that are executable in parallel. The function recited in claim 1(a) is therefore performed identically.

52. The compiler utilized by the accused processors is the same or equivalent structure disclosed in the '003 patent specification for performing this function.

53. Each of the accused processors includes a super-scaler cache as defined by the court for routing the instructions in the groups to appropriate instruction pipelines of a plurality of instruction pipelines.

54. Each of the accused processors includes a super-scaler storage for holding a group of instructions. The rotate buffer found in the accused processors is a super-scaler storage.

55. Each of the accused processors has an associative crossbar, as described above.

56. The first set of connectors is coupled to the rotate buffer for receiving each of the software-scheduled instructions therefrom.

57. The second set of connectors couples the crossbar to the processing pipelines.

58. Each of the accused processors includes circuitry for performing the function of coupling appropriate connectors of the first set of connectors to appropriate connectors of the second set of

connectors, to thereby supply each of the software-scheduled instructions to the appropriate instruction pipeline for parallel execution.

59. The accused processors utilize structure that is the same as or equivalent to the structures in the '003 patent specification that perform the identical function identified in the preceding finding of fact. Tr. Trans. 7/3/02 a.m. @ 70-74. Those structures in the accused processors receive the pipeline identifiers and respond to the pipeline identifiers to perform the recited function.

60. The accused processors contain all of the limitations of claim 1 of the '003 patent.

61. Independent claim 6 provides:

> 6. In a computing system, a method for transferring software-scheduled instructions to be executed through an associative crossbar switch in a super-scaler cache, the associative crossbar switch having a first set of connectors coupled to a super-scaler storage in the super-scaler cache for receiving each of the software-scheduled instructions therefrom and a second set of connectors coupled to a plurality of instruction pipelines, the method comprising the steps of:
>
> forming groups of software-scheduled instructions, software-scheduled instructions within each group executable in parallel;
>
> storing in the super-scaler storage in the super-scaler cache one group of the groups of software-scheduled instructions to be executed in parallel, each software-scheduled instruction in the one group having embedded therein an instruction pipeline identifier of a plurality of instruction pipeline identifiers; and
>
> using the instruction pipeline identifier of each of the software-scheduled instructions to control switches in the associative crossbar switch in the super-scaler cache between the first set of connectors and the second set of connectors to thereby supply each of the software-scheduled instructions to an appropriate instruction pipeline.

62. The accused processors perform a method of transferring instructions through an associative crossbar.

63. The associative crossbar has a first set of connectors coupled to a line in the I-cache, as well

as the rotate buffer and a second set of connectors coupled to a plurality of processing pipelines.

64. The accused processors use a compiler to form groups of instructions executable in parallel.

65. The instructions are software-scheduled.

66. The accused processors store in a line of the I-cache, as well as in the rotate buffer, one group of instructions, and each of the instructions has embedded therein a pipeline identifier of a plurality of instruction pipeline identifiers for the reasons expressed previously in the court's findings of fact related to claim 20 of the '028 patent.

67. The pipeline identifiers in the template information control the steering crossbar (Itanium) and the multiplexers (Itanium 2) to supply each individual instruction to the appropriate processing pipeline.

68. Therefore, all limitations of claim 6 of the '003 patent are found in the accused processors.

69. Claim 7 depends from claim 6.

70. Claim 7 provides:

> The method of claim 6 wherein the step of using the instruction pipeline identifier comprises:
>
> supplying the instruction pipeline identifiers of each of the software-scheduled instructions in the one group of software-scheduled instructions to a corresponding number of selectors, each of the selectors providing an output signal indicative of the instruction pipeline identifier; and
>
> using the output signal of each of the selectors to control the switches between the first set of connectors and the second set of connectors to thereby supply each of the software-scheduled instructions to the appropriate instruction pipeline.

71. For the reasons expressed in the court's findings of fact related to claim 21 of the '028 patent, which reasons are incorporated by reference, the court finds that the accused processors meet each limitation of claim 7 of the '003 patent.

72. Intergraph also asserts claim 8 of the '003 patent. From the evidence found to be persuasive, the court does not find by a preponderance of the evidence that the accused devices meet at least the final element of claim 8 that requires "a switch coupled to the associative crossbar, and to the selector, for asserting switch selection signals to the associative crossbar in response to the output signals." Therefore, the accused devices do not meet all of the limitations of claim 8 of the '003 patent.

73. Claim 16 depends from claim 8.

74. For the reasons expressed previously, the accused devices do not meet all of the limitations of claim 16 of the '003 patent.

75. Claim 11 depends from claim 1. Claim 11 provides "[t]he computing system of claim 1, wherein the means for forming the groups of software-scheduled instructions is determined by a compiler."

76. The accused processors meet the additional limitations of claim 11 because the accused processors utilize a means for forming the groups of software-scheduled instructions that is determined by a compiler. In the accused processors, the compiler determines the formation of the groups of software-scheduled instructions. Therefore, all limitations of claim 11 are found in the accused processors.

77. Claim 14 depends from claim 6. For the reasons expressed in the court's findings of fact related to claim 11 of the '003 patent, the court finds that the accused processors meet each limitation of dependent claim 14 because the step of forming groups of software-scheduled instructions comprises using a compiler.

78. Claim 15 depends from claim 7. For the reasons expressed in the court's findings of fact

related to claim 11 of the '003 patent, the court finds that the accused processors meet each limitation of dependent claim 15 because the instruction pipeline identifiers are determined by a compiler.

79. Independent claim 22 provides:

> 22. In a computing system in which sets of software-scheduled instructions are executable in parallel, a super-scaler cache for routing each of the software-scheduled instructions in a group to be executed in parallel, to an appropriate instruction pipeline of a plurality of instruction pipelines, the super-scaler cache comprising:
>
> a super-scaler storage for holding the sets of software-scheduled instructions, including at least a set of software-scheduled instructions, and a set of instruction pipeline identifiers, each individual instruction of the first set of software-scheduled instructions having associated therewith an instruction pipeline identifier of the set of instruction pipeline identifiers;
>
> an associative crossbar having a first set of connectors coupled to the super-scaler storage for receiving the set of software-scheduled instructions therefrom and a second set of connectors coupled to the plurality of instruction pipelines;
>
> selection means connected to receive the instruction pipeline identifiers of the set of instruction pipeline identifiers, the selection means for supplying in response thereto output signals; and
>
> switching means coupled to receive the output signals for selectively connecting connectors of the first set of connectors to connectors of the second set of connectors to thereby supply each software-scheduled instruction in the set of software-scheduled instructions to be executed in parallel to the appropriate instruction pipeline;
>
> wherein the instruction pipeline identifiers of the set of instruction pipeline identifiers are determined by a compiler.

80. For the reasons expressed in the court's findings of fact related to claim 1 of the '003 patent and the additional reasons expressed herein, the court finds that all limitations of claim 22 are found in the accused processors.

81. The accused processors have a selection means connected to receive the instruction pipeline identifiers of the set of instruction pipeline identifiers, the selection means for supplying in response thereto output signals.

82. The accused processors use dispersal control logic to control the multiplexers in the accused processors. The accused processors thus perform the identical function as recited in the claims and use structures that are the same as or equivalent to those disclosed in the patent specification that perform the function.

83. The accused processors also have a switching means coupled to receive the output signals for selectively connecting connectors of the first set of connectors to connectors of the second set of connectors to thereby supply each software-scheduled instruction in the set of software-scheduled instructions to be executed in parallel to the appropriate instruction pipeline.

84. The steering crossbar in the Itanium processor and the multiplexers in the Itanium 2 process are the switching means coupled to receive the output signals for selectively connecting connectors of the first set of connectors to connectors of the second set of connectors to thereby supply each software-scheduled instruction in the set of software-scheduled instructions to be executed in parallel to the appropriate instruction pipeline. Therefore, the accused processors perform the identical function recited in this limitation of claim 22 and use the same or equivalent structure to do so.

85. The accused processors meet the final limitation of claim 22 because a compiler determines the instruction pipeline identifiers of the set of instruction pipeline identifiers.

86. The accused processors meet all of the limitations of claim 22 of the '003 patent.

87. For the reasons expressed in connection with claim 8, the evidence found to be persuasive does not convince the court that the plaintiffs have proved the existence of all of the limitations of

independent claim 28. In particular, the plaintiffs have not proved at least one of the limitations of claim 28–specifically, that the accused processors have a switch coupled to the associative crossbar, and to the selector, for asserting switch selection signals to the associative crossbar in response to the output signals.

88. The accused processors do not meet all of the limitations of claim 28 of the '003 patent.

Invalidity Analysis

89. Intel has failed to persuade the court by clear and convincing evidence that any claims of the '028 or the '003 patent are invalid as being anticipated by the prior art references.

90. Intel has failed to persuade the court by clear and convincing evidence that any of the claims of the '028 patent or the '003 patent are invalid as being rendered obvious by the prior art references.

91. Intel has failed to persuade the court by clear and convincing evidence that any of the claims of the '028 patent or the '003 patent that require the presence of a crossbar are invalid as anticipated by either of the two Vassiliadis Papers.

92. As to claims 6-9 of the '028 patent, Intel has failed to persuade the court by clear and convincing evidence that claims 6-9 of the '028 patent are invalid as anticipated by either of the two Vassiliadis Papers. The evidence does not support a finding, by clear and convincing evidence, that the very long instruction word is stored in main memory.

CONCLUSIONS OF LAW

1. This court has exclusive subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338.

2. The court has personal jurisdiction over the defendant in this action.

3. Venue is proper in this district pursuant to 28 U.S.C. § 1400.

4. The court incorporates by reference the canons of claim construction recited in its June 3, 2002 claim construction order. The court applies those canons, together with the additional rules expressed in these Findings of Fact and Conclusion of Law to construe the meaning of the term "instruction."

5. The court has considered carefully the language of the claims, the specification and the file history and is persuaded that the term "instruction" as used in the '028 and '003 patents means "the smallest unit of work capable of being processed at any particular stage in the computer."

6. Claim construction begins with the language of the claims themselves. In the claims of the patents- in-suit, the court should ordinarily give the word "instruction" the same meaning it would have to one skilled in the particular art. The court must review the specification and the file history to determine whether the inventor gave the term a different or more limited meaning that the term would ordinarily have.

7. In this case, Intel asserts that the term "instruction" as used in the patents, means "a traditional programmer visible command, including a full opcode, that uniquely describes one basic computer command." Intel relies on portions of Dr. Peuto's testimony that a syllable is not enough to be an instruction and that an instruction is a programmer visible instruction. According to Intel, the court's adoption of a contrary definition would violate the canon of claim construction that requires the construction of terms consistently throughout the patents. *See Digital Biometrics, Inc. v. Identix, Inc.*, 149 F.3d 1335, 1345 (Fed. Cir. 1998)("the same word appearing in the same claim should be interpreted consistently"); *Fonar Corp. v. Johnson & Johnson*, 821 F.2d 627, 632 (Fed. Cir. 1987)(the meaning of a term in a claim must be defined in a manner that is consistent with its appearance in other claims in the same patent).

8. The court disagrees with Intel's construction because the construction is inconsistent with the patents' specifications. In the context of the Intergraph patents, an "instruction" assumes different forms depending on the stage of processing. For instance, during compilation an instruction may not include group or pipeline identifiers. '028 patent at 8:6-13; 6:13-14; 7:36-38. Likewise, in one embodiment, during pre-decoding, the form of the instruction changes from 32 bits to 64 bits. '28 patent at 4:58-60; 6:10-12; 7:56-58; 8:35-37.

9. Intel overstates the effect of the court's decision on the canon of claim construction as it applies to this case. While it is true that, under the court's construction, what constitutes an instruction varies depending on the location of the instruction in its processing, the court's definition of instruction is the same throughout the claims. The court's definition is therefore consistent with Intel's canon of construction.

10. The court's construction of instruction is also consistent with what one skilled in the art would believe it to be. *E.g.*, Crawford Testimony, Trial Transcript, July 9, 2002 a.m., p. 64-65 ("an instruction is the smallest unit of work that a computer can carry out."). The term "instruction" as used in the '028 and '003 patents means "the smallest unit of work capable of being processed at any particular stage in the computer."

11. On infringement, the patent statute provides:

> Except as otherwise provided in this title, whoever without authority makes, uses, offers to sell or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent. 35 U.S.C. § 271(a).

12. Infringement requires proof by a preponderance of the evidence. *Kegel Co. v. AMF Bowling, Inc.*, 127 F.3d 1420, 1425 (Fed. Cir. 1997). To determine infringement, the court performs a two

step analysis. First, the court construes the claims to define their scope. *Cybor Corp. v. FAS Techs., Inc.*, 138 F.3d 1448, 1456 (Fed. Cir. 1998)(en banc). Second, the court must, as a finder of fact, examine the evidence to determine whether the accused product infringes the properly construed claims. *Kegel*, 127 F.3d at 1425.

13. Each and every element in a patent claim must be found in an accused device to find infringement. *Glaxo, Inc. v. Novopharm, Ltd.*, 110 F.3d 1562, 1566 (Fed. Cir. 1997); *Cortland Line Co. v. Orvis Co.*, 203 F.3d 1351, 1358 (Fed. Cir. 2000)(literal infringement requires "that the properly construed claim reads on the device exactly.").

14. For literal infringement of a means-plus-function element, the second step of the infringement analysis begins with determining whether the accused device or method performs an identical function to the one recited in the claim. If the identical function is performed, the next step is to determine whether the accused device uses the same structure, materials, or acts found in the specification, or their equivalents. *IMS Technology, Inc. v. Haas Automation, Inc.*, 206 F.3d1422, 1430 (Fed. Cir. 2000). "To determine whether a claim limitation is met literally, where expressed as a means for performing a stated function, the court must compare the accused structure with the disclosed structure, and must find equivalent structure as well as identity of claimed function for that structure." *Pennwalt Corp. v. Durand-Wayland, Inc.*, 833 F.2d 931, 934 (Fed. Cir. 1987)(en banc).

15. "An accused device that does not literally infringe a claim may still infringe under the doctrine of equivalents if each limitation of the claim is met in the accused device either literally or equivalently." *Cybor*, 138 F.3d at 1459. A means-plus-function element may infringe under the doctrine of equivalents without identity of function if the function performed by the accused device is substantially the same as that claimed. *WMS Gaming Inc. v. International Game Technology*, 184

F.3d 1339, 1353 (Fed. Cir. 1999).

16. Intergraph has not proven infringement of any of the asserted claims under either of the patents in suit by the doctrine of equivalents. Therefore, the court is only concerned with whether Intergraph has proven literal infringement of any of the asserted claims of either of the patents in suit.

17. The accused devices do not literally infringe independent claims 6, 11 and 16 of the '028 patent because Intergraph has failed to persuade the court that the accused devices contain the limitation of a group identifier. Necessarily, the accused devices do not literally infringe any claims that depend from claims 6, 11 and 16. Therefore, the accused products do not literally infringe either claim 7 or claim 9, both of which depend from claim 6.

18. The accused devices literally infringe claim 20 of the '028 patent because each and every limitation in claim 20 is found in the accused devices.

19. The accused devices literally infringe claim 21 of the '028 patent because each and every limitation in claim 21 is found in the accused devices.

20. The accused devices literally infringe claim 1, 6, 7, 11, 14, 15 and 22 of the '003 patent because each and every limitation of those claims is found in the accused devices.

21. A patent is presumed valid, and the burden is on the challenger of the validity of a patent to prove by clear and convincing evidence that the patent is invalid. *Applied Materials v. Advanced Semiconductor Materials*, 98 F.3d 1563, 1569 (Fed. Cir. 1996).

22. Title 35 U.S.C. section 102(a) provides that a person shall be entitled to a patent unless "the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent." An invention is not novel when it is "anticipated" by a prior art reference. "The classic test of

anticipation provides: 'That which will infringe, if later, will anticipate, if earlier." Thus, a claim fails to meet the novelty requirement if it covers or reads on a product or process found in a single source in the prior art." 1 Chisum on Patents § 3.02[1], pp. 3-12 to 3-14 (1998). The party asserting that patent claims are invalid under 35 U.S.C. § 102(a) for lack of novelty must show by clear and convincing evidence that each element of a claim in issue is contained in a single prior art reference. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 771 (Fed. Cir. 1983).

23. Intel has not persuaded the court by clear and convincing evidence that any of the claims of the '028 patent or the '003 patent are invalid based on anticipation.

24. The '028 patent is valid and enforceable.

25. The '003 patent is valid and enforceable.

26. A court "may grant injunctions in accordance with principles of equity to prevent the violation of any right secured by patent on such terms as the court deems reasonable. 35 U.S.C. § 283. The court must consider all of the circumstances, including the adequacy of the legal remedy, irreparable injury, whether the public interest would be served, and the hardship on the parties and third parties. *Roche Prod. v. Bolar Pharmaceutical Co.*, 733 F.2d 1230, 1233 (Fed. Cir. 1984); *Smith Int'l, Inc. v. Hughes Tool Co.*, 718 F.2d 1573, 1577-79 (Fed. Cir. 1983).

27. In patent cases, once validity and infringement are established, there is a presumption of immediate irreparable harm. *Smith Int'l*, 718 F.2d at 1578.

28. In patent cases, where the infringing device will continue to infringe, monetary damages are generally considered to be inadequate. *Atlas Powder Co. v. Ireco Chem.*, 733 F.2d 1230, 1233 (Fed. Cir. 1985); *Shiley, Inc. v. Bentley Laboratories*, 601 F.Supp. 964, 970 (C.D. Cal. 1985).

29. The public interest factor favors the patentee, given the public's interest in maintaining the

integrity of the patent system. *Odetics, Inc. v. Storage Technology Corp.*, 14 F. Supp. 2d 785, 795 (E.D. Va. 1998).

30. Intergraph is entitled to an injunction preventing Intel from manufacturing, using, selling, offering to sell, or importing into the United States Intel's Itanium or Itanium 2 processors.

31. Intergraph is directed to submit a proposed form of judgment and injunction within ten (10) days from the entry of these Findings of Fact and Conclusions of Law.

So **ORDERED** and **SIGNED** this 10th day of October, 2002.

T. John Ward

T. JOHN WARD
UNITED STATES DISTRICT JUDGE